



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

10/731,281

12/09/2003

Jianbing Huang

05-03-014

4973

45113 7590  
DOCKET CLERK  
PO BOX 800889  
DALLAS, TX 75380

08/07/2007

EXAMINER

NGUYEN, KIMBINH T

ART UNIT

PAPER NUMBER

2628

MAIL DATE

DELIVERY MODE

08/07/2007

PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.



UNITED STATES PATENT AND TRADEMARK OFFICE

---

Commissioner for Patents  
United States Patent and Trademark Office  
P.O. Box 1450  
Alexandria, VA 22313-1450  
[www.uspto.gov](http://www.uspto.gov)

**BEFORE THE BOARD OF PATENT APPEALS  
AND INTERFERENCES**

Application Number: 10/731,281  
Filing Date: December 09, 2003  
Appellant(s): HUANG ET AL.

**MAILED**

**AUG 07 2007**

**Technology Center 2600**

---

Mathew S. Anderson  
For Appellant

**EXAMINER'S ANSWER**

This is in response to the appeal brief filed 01/16/2007 appealing from the Office action mailed August 11, 2006.

**(1) Real Party in Interest**

A statement identifying by name the real party in interest is contained in the brief.

**(2) Related Appeals and Interferences**

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

**(3) Status of Claims**

The statement of the status of claims contained in the brief is correct.

**(4) Status of Amendments After Final**

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

**(5) Summary of Claimed Subject Matter**

The summary of claimed subject matter contained in the brief is correct.

**(6) Grounds of Rejection to be Reviewed on Appeal**

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

**(7) Claims Appendix**

The copy of the appealed claims contained in the Appendix to the brief is correct.

**(8) Evidence Relied Upon**

The prior art of record:

Art Unit: 2628

6,912,293	KOROBKIN	6-1999
20040114794	VLASIC et al.	12-2002

### **(9) Grounds of Rejection**

The following ground(s) of rejection are applicable to the appealed claims:

#### ***Claim Rejections - 35 USC § 103***

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

1. Claims 1, 2, 4, 5, 8-12, 14, 15, 18-22, 24, 25 and 28-30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Korobkin (6,912,293) in view of Brokenshire et al. (6,624,810).

**Claim 1**, Korobkin discloses receiving node and view data (in a 3D graphics API such as OpenGL or direct 3D; col. 16, lines 63-65) for a graphic object (col. 17, lines 30-37); building a binary-space-partition tree corresponding to the graphic object (col. 15, lines 32-48), the binary-space-partition tree having up to a predetermined number of at least one shape associated with each leaf (G-regions and U-regions appear as leaves in the tree. G-regions are tagged to identify their specific origin. Insertion of geometry stops when there are no U-region nodes (screen is full); col. 15, lines 64-67); sorting shapes at each leaf of the binary-space-partition tree (col. 15, lines 33-36); and

Art Unit: 2628

outputting the sorted shapes (a subsequent traversal of the tree will deliver triangles (shapes) in a spatial correct "back-to-front"; col. 15, lines 37-38). Korobkin does not clearly teach binary space partition tree having up to a predetermined number of at least one shape associated with each leaf; but Korobkin implicitly discloses that U-regions as leafs. Insertion of geometry stops; it means the partition having up to a predetermined number and stops the partition; as one of ordinary skill in the art, the partition of the binary space tree should be continued until a desired number of levels of subspace or sub-object has been created and stopped the partition; however, Brokenshire et al. explicitly teaches this feature (Fig. 3A: The plane A is the root of the tree and its two sub-planes B and C are the children of node A; col. 4, lines 33-61; col. 5, lines 51-59; col. 6, lines 2-9; at each level, each subspace is partitioned into further subspaces until a desired number of levels of subspaces has been created. For 3D, each subspace (leaf) may be a rectangular solid or some other polyhedron; col. 6, lines 53-65; once the space has been subdivided into a predetermined number of level of subspaces; col. 7, lines 3-9). It is noted that Korobkin and Brokenshire teach partition tree in both object-space and image-space with hyper-planes (Korobkin at col. 14, lines 35-54, col. 15, lines 32-38 and Brokenshire at col. 4, lines 33-54). Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the method of binary-space partitioning tree of Brokenshire into the BSP tree of Korobkin to accomplish a global visibility sort of the input database (Korobkin; col. 15, lines 22-23), because once the space has been subdivided into a predetermined number of level of subspaces, the bounding volume of each space or

subspace is recomputed such that the bounding volume just contains the object or objects that fit into that level of BSP tree. Thus the dead space at each level of the BSP tree is reduced by this recalculation of the bounding volume. This recalculation of the bounding volume may be accomplished by analyzing the values of the primitives that define the object, for which an image will be rendered and displayed to a user (Brokenshire, col. 7, lines 3-14).

**Claim 2**, Korobkin discloses the shapes are sorted into a substantially back-to-front order (col. 15, lines 34-38).

**Claim 4**, Korobkin discloses traversing the binary-space-partition tree (col. 15, lines 36-38; col. 31, lines 43-51).

**Claim 5**, Korobkin discloses the shapes are triangles (col. 15, line 37; figs 5a).

**Claim 8**, Korobkin discloses analyzing shapes (triangles) in a graphic object (fig. 15); creating a root node and a list of additional nodes for a binary-space-partition tree (col. 17, lines 56-62), each node associated with at least one shape (col. 19, lines 6-16); performing a partition plane selection for each additional node, classifying the shapes at the additional node according to the partition plane selection (col. 15, lines 51-63); and creating child nodes according to the shape classification (leaf nodes; col. 15, lines 64-66).

**Claim 9**, Korobkin discloses each node represents a set of elements located in a 3-D spatial region (col. 19, lines 45-51; figs. 9a, 9b).

**Claim 10**, the rationale provided in the rejection of claim 5 is incorporated herein.

**Claims 11, 12, 14, 15, 18-20** the rationale provided in the rejection of claims 1, 2, 4, 5, 8-10 are incorporated herein. Specially, independent claim 18, Korobkin teaches analyzing shapes in a graphic object (subdivides the object space and geometry (triangle) with hyper-planes defined by the surface facet of the input; col. 15, lines 32-36); creating a root nodes and a list of additional nodes for a binary-space partition (figs. 8d, 9b, 29b, 37, 38b), each node associated with up to a predetermined number of at least one shape or element or triangle; as one of ordinary skill in the art, the partition of the binary space tree either in object-space or image-space should be continued until a desired or predetermined number of levels of subspace or sub-object (leaf or child) has been created and stopped the partition for a valid scene graph database; however, Brokenshire et al. explicitly teaches this feature (Fig. 3A: The plane A is the root of the tree and its two sub-planes B and C are the children of node A; sub-planes D and E are polytopes or elements or shape associated with leaf or child element; col. 4, lines 33-61; col. 5, lines 51-59; col. 6, lines 2-9; at each level, each subspace is partitioned into further subspaces until a desired number of levels of subspaces has been created. For 3D, each subspace (leaf) may be a rectangular solid or some other polyhedron; col. 6, lines 53-65; once the space has been subdivided into a predetermined number of level of subspaces; col. 7, lines 3-9). It is noted that Korobkin and Brokenshire teach partition tree in both object-space and image-space with hyper-planes (Korobkin at col. 14, lines 35-54, col. 15, lines 32-38; figs. 9A, 9B the leaf node for final partition is the shape of "arch" and Brokenshire at col. 4, lines 33-54; figs. 3A, 6A, sub-planes D and E are the leaf nodes); Korobkin teaches performing a partition plane selection for each additional

node classifying the shape at additional nodes, creating child nodes according to the shape classification (fig. 9b, the additional node "door" partitioned into "arch"; fig. 37: "left-wall " node partitioned into "fireplace"). It would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the method of binary-space partitioning tree of Brokenshire into the BSP tree of Korobkin to accomplish a global visibility sort of the input database (Korobkin; col. 15, lines 22-23), because once the space has been subdivided into a predetermined number of level of subspaces, the bounding volume of each space or subspace is recomputed such that the bounding volume just contains the object or objects that fit into that level of BSP tree. Thus the dead space at each level of the BSP tree is reduced by this recalculation of the bounding volume. This recalculation of the bounding volume may be accomplished by analyzing the values of the primitives that define the object, for which an image will be rendered and displayed to a user (Brokenshire, col. 7, lines 3-14).

**Claims 21, 22, 24, 25 and 28-30** the rationale provided in the rejection of claims 1, 2, 4, 5, 8-10 are incorporated herein. In addition, Korobkin teaches a computer program product tangibly embodied in a machine-readable medium (col. 7, lines 6-50).

2. Claims 3, 6, 7, 13, 16, 17, 23, 26 and 27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Korobkin (6,912,293) in view of Brokenshire et al. (6,624,810) and further in view of Vlasic et al. US 2004/0114794.

**Claims 3, 6 and 7**, Korobkin does not teach caching the shape data; Vlasic et al. discloses caching the shape data (paragraphs 0050, 0055 and 0056); a configuration component is used (the graphic hardware 141), the configuration component balancing



the resolution of the binary-space-partition tree against the sorting shapes at each leaf (paragraph 0074); a configuration component is used (graphics hardware), the configuration component balancing resource usage against accuracy in the resolution of the caching (the alpha blending is incorrect (paragraphs 0059, 0060). It would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the conventional graphics hardware, caching the projective texture coordinates of the shape data taught by Vlasic into the system constructing 3D scene of Korobkin to accomplish a global visibility sort of the input database, because it would provide high quality and high performance rendering (paragraph 0076).

**Claims 13, 16, 17**, the rationale provided in the rejection of claims 3, 6 and 7 are incorporated herein.

**Claims 23, 26, 27**, the rationale provided in the rejection of claims 3, 6, 7 and 21 are incorporated herein.

#### **(10) Response to Argument**

With respect to applicant's arguments claim 1, Korobkin implicitly discloses that G-regions and U-regions as leaf nodes in the tree. Insertion of geometry stops; it means the partition having up to a predetermined number and stops the partition; as one of ordinary skill in the art, the partition of the binary space tree either in object-space or image-space should be continued until a desired or predetermined number of levels of subspace or sub-object (leaf or child) has been created and stopped the partition for a valid scene graph database; however, Brokenshire et al. explicitly teaches this feature (Fig. 3A: The plane A is the root of the tree and its two sub-planes B and C are the

children of node A; sub-planes D and E are polytopes or elements or shape associated with leaf or child element; col. 4, lines 33-61; col. 5, lines 51-59; col. 6, lines 2-9; at each level, each subspace is partitioned into further subspaces until a desired number of levels of subspaces has been created. For 3D, each subspace (leaf) may be a rectangular solid or some other polyhedron; col. 6, lines 53-65; once the space has been subdivided into a predetermined number of level of subspaces; col. 7, lines 3-9). It is noted that Korobkin and Brokenshire teach partition tree in both object-space and image-space with hyper-planes (Korobkin at col. 14, lines 35-54, col. 15, lines 32-38; figs. 9A, 9B the leaf node for final partition is the shape of "arch" and Brokenshire at col. 4, lines 33-54; figs. 3A, 6A, sub-planes D and E are the leaf nodes). And the combination of Korobkin and Brokenshire teachings are proper because both of them teach the methods of binary-space partition with hyper-plane to implement graphical database for which an image will be rendered and displayed to a user.

Claim 2 depends from claim 1 and the limitation as disclosed by Korobkin.

Claim 4 depends from claim 1 and the discussion of claim 1 above is applied hereto.

Claim 5 depends from claim 1 and the discussion of claim 1 above is applied hereto.

Claim 8, the discussion of claim 1 above is incorporated hereto. As one of ordinary skill in the art, the partition of the binary space tree either in object-space or image-space should be continued until a desired or predetermined number of levels of

sub-space or sub-object (leaf or child) has been created and stopped the partition or insertion as disclosed by Korobkin (col. 15, lines 64-67).

Claim 9 depends from claim 8, the discussion above with regard to claim 8 apply hereto. In addition, Korobkin discloses each node represents a set of elements located in a 3-D spatial region (col. 19, lines 45-51; figs. 9a, 9b).

Claim 10 depends from claim 8, the discussion above with regard to claim 8 apply hereto.

Claim 11, the rationale provided in the rejection of claim 1 above is incorporated herein. In addition, Korobkin teaches sub-triangles, sub-planes, U-regions and R-regions appear as leaf nodes in the tree. Insertion of geometry stops when there are no U-region nodes (screen is full), (Korobkin, col. 15, lines 32-67), it means "some pre-set condition is met" (paragraph 0043 of the application) or as described in the Brokenshire's reference "once the space has been subdivided into a predetermined number of levels of subspaces, the bounding volume of each space or subspace is recomputed such that the bounding volume just contains the object or objects that fit into that level of the BSP tree", (Brokenshire, col. 7, lines 3-7).

Claim 12, the discussion above with regard to claim 11 is applied here as well, and are incorporated herein by references.

Claim 14, the discussion above with regard to claim 11 is applied here as well, and are incorporated herein by references.

Claim 15, the discussion above with regard to claim 11 is applied here as well, and are incorporated herein by references.

Claim 18, Korobkin teaches analyzing shapes in a graphic object (subdivides the object space and geometry (triangle) with hyper-planes defined by the surface facet of the input; col. 15, lines 32-36); creating a root nodes and a list of additional nodes for a binary-space partition (figs. 8d, 9b, 29b, 37, 38b), each node associated with up to a predetermined number of at least one shape or element or triangle; as one of ordinary skill in the art, the partition of the binary space tree either in object-space or image-space should be continued until a desired or predetermined number of levels of subspace or sub-object (leaf or child) has been created and stopped the partition for a valid scene graph database; however, Brokenshire et al. explicitly teaches this feature (Fig. 3A: The plane A is the root of the tree and its two sub-planes B and C are the children of node A; sub-planes D and E are polytopes or elements or shape associated with leaf or child element; col. 4, lines 33-61; col. 5, lines 51-59; col. 6, lines 2-9; at each level, each subspace is partitioned into further subspaces until a desired number of levels of subspaces has been created. For 3D, each subspace (leaf) may be a rectangular solid or some other polyhedron; col. 6, lines 53-65; once the space has been subdivided into a predetermined number of level of subspaces; col. 7, lines 3-9). It is noted that Korobkin and Brokenshire teach partition tree in both object-space and image-space with hyper-planes (Korobkin at col. 14, lines 35-54, col. 15, lines 32-38; figs. 9A, 9B the leaf node for final partition is the shape of "arch" and Brokenshire at col. 4, lines 33-54; figs. 3A, 6A, sub-planes D and E are the leaf nodes); Korobkin teaches performing a partition plane selection for each additional node classifying the shape at additional nodes, creating child nodes according to the shape classification (fig. 9b, the

additional node "door" partitioned into "arch"; fig. 37: "left-wall " node partitioned into "fireplace"). It would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the method of binary-space partitioning tree of Brokenshire into the BSP tree of Korobkin to accomplish a global visibility sort of the input database (Korobkin; col. 15, lines 22-23), because once the space has been subdivided into a predetermined number of level of subspaces, the bounding volume of each space or subspace is recomputed such that the bounding volume just contains the object or objects that fit into that level of BSP tree. Thus the dead space at each level of the BSP tree is reduced by this recalculation of the bounding volume. This recalculation of the bounding volume may be accomplished by analyzing the values of the primitives that define the object, for which an image will be rendered and displayed to a user (Brokenshire, col. 7, lines 3-14).

Claim 19, the discussion above with regard to claim 18 is applied here as well, and are incorporated herein by references.

Claim 20, the discussion above with regard to claim 18 is applied here as well, and are incorporated herein by references.

Claim 21 claims a computer program product tangibly embodied in a machine readable medium for performing the steps of claim 1; therefore it is rejected under the same reason set forth in claim 1.

Claim 22 depends upon claim 21, the discussion above with regard to claims 1 and 21 apply here as well, and are incorporated herein by the reference.

Claim 24 depends upon claim 21, the discussion above with regard to claims 1 and 21 apply here as well, and are incorporated herein by the reference.

Claim 25 depends upon claim 21, the discussion above with regard to claims 1 and 21 apply here as well, and are incorporated herein by the reference.

Claim 28 claims a computer program product tangibly embodied in a machine readable medium for performing the steps of claim 18; the discussion above with regard to claim 18 apply here as well, and are incorporated herein by the reference.

Claim 29 depends from claim 28, the discussion above with regard to claims 18 and 28 apply here as well, and are incorporated herein by the reference.

Claim 30 depends from claim 28, the discussion above with regard to claims 18 and 28 apply here as well, and are incorporated herein by the reference.

Claim 3 depends from claim 1, the discussion above with regard to claim 1 is incorporated herein. In addition, claim 3 further comprising caching the shape data as taught by Vlasic's reference.

Claim 6 depends from claim 1, the discussion above with regard to claim 1 is incorporated herein. In addition, claim 6 further limitation as taught by Vlasic's reference (balancing the resolution of the binary-space tree by: traversal, resizing texture, caching texture coordinates, combining weights, hardware shaders, and rendering, paragraph 0074).

Claim 7 depends from claim 3, the discussion above with regard to claims 1 and 3 are incorporated herein. In addition, Vlasic teaches each triangle (resource usage) must be rendered in a single pass. If there were more than one pass per triangle , then

Art Unit: 2628

the alpha blending (the resolution of the caching coordinate data) is incorreced (paragraph 0060).

Claim 13 depends from claim 11, the discussion above with regard to claim 11 is incorporated herein. In addition, Vlastic teaches caching the shape data (caching data structure; paragraph 0050).

Claim 16, the discussion above with regard to claim 6 is incorporated herein.

Claim 17, the discussion above with regard to claim 7 is incorporated herein.

Claim 23, the discussion above with regard to claim 3 is incorporated herein.

Claim 26, the discussion above with regard to claim 6 is incorporated herein.

Claim 27, the discussion above with regard to claim 7 is incorporated herein.

The motivation to combine Korobkin with Brokenshire and Vlastic are taught and disclosed by the art of record (see the detailed rejection of the claims) and proper obviousness rejection. It would have been obvious to one of ordinary skill in the art at the time the invention was made to utilize a binary-space partition for reducing data structure, each space or object is subdivided or partitioned by a hyperplane until a desired number of levels of subspace or subplane has been created (predetermined number) and the final node of the partition should be a leaf or a child node which associated with the element or space or region.

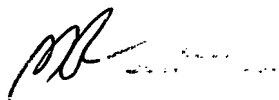
#### **(11) Related Proceeding(s) Appendix**

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

Art Unit: 2628

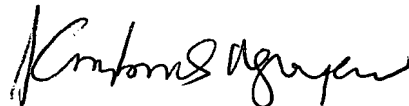
For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted;



Michael Razavi

May 17, 2007



**KIMBINH T. NGUYEN**  
**PRIMARY EXAMINER**